Tartare: Adding RawMeat Functionality to the Xcalibur Workbench

Michael W. Senko, Derek J. Bailey, Michael P. Goodwin, and Romain Huguet, Thermo Fisher Scientific, San Jose, CA, USA, 95134

OVERVIEW

- **Tartare provides quality assessment tools for data dependent experiments**
- **Tartare extends prior RawMeat capabilities**
- **Tartare supports all current Xcalibur-based instruments**
- **Tartare supports user-written analysis scripts**

INTRODUCTION

Data dependent acquisitions on modern hybrid mass spectrometers can generate tens to hundreds of thousands of spectra in a single acquisition. For bottom-up proteomics, data quality is usually assessed based on the number of Peptide Spectral Matches (PSM's). Due to the vast amount of raw data, it's difficult to relate PSM counts back to any particular aspect of the original acquisitions. RawMeat (VAST Scientific) is a free, easy-to-use assessment tool for evaluating the quality of data dependent experiments, including tabular and graphical representations of precursor information such as charge state, intensity and injection times.

RawMeat has not been updated in many years and is no longer fully compatible with data files generated by modern instruments. Here we describe the addition of new quality assessment features into the Thermo Scientific™ Xcalibur™ Workbench. We call this RawMeat replacement *Tartare*.

METHODS

Tartare is implemented as part of the Xcalibur Workbench[1]. The Xcalibur Workbench (XWB) is written entirely in Lua and uses various libraries to support the GUI and access Xcalibur raw files (https://github.com/thermofisherlsms/xcalibur-workbench). XWB is compatible with the standard Lua 5.1 interpreter as well as specialty flavors of LuaJIT, such as GSL Shell and Torch.

The framework for Tartare consists of ~600 lines of Lua code which provides functionality such as raw file access and common plotting routines to support the individual analysis. Each analysis consisting of 30-200 lines of additional code, retrieves specific information from the raw file and displays the results in an appropriate format.

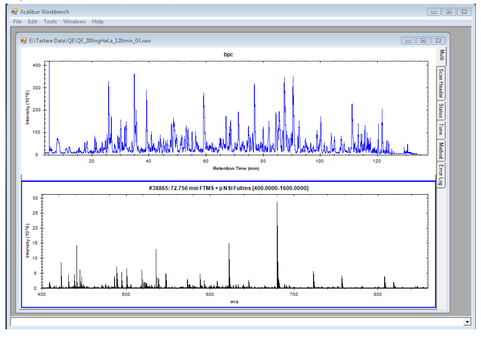
Although written entirely in a scripting language, Tartare is significantly faster than RawMeat, with processing taking ~30 seconds per gigabyte of data. Tartare has been tested with data from various generations of Xcalibur-based mass spectrometers, such as the Thermo Scientific™ LTQ™ FT, Thermo Scientific™ LTQ™ Orbitrap Elite™, Thermo Scientific™ Q Exactive™ and Thermo Scientific™ Orbitrap Fusion™ Tribrid™ mass spectrometers.

RESULTS

Xcalibur Workbench

The Xcalibur Workbench (Figure 1) is a data browser which provides general support for mass spectral data analysis, with direct access to Xcalibur data files and powerful plotting capabilities supported through ZedGraph (http://zedgraph.sourceforge.net/samples.html). Several of the features of RawMeat are available as a standard part of XWB, such as display of chromatograms, method reports, and error logs.

Figure 1. Xcalibur Workbench GUI.



Basic Features of Tartare

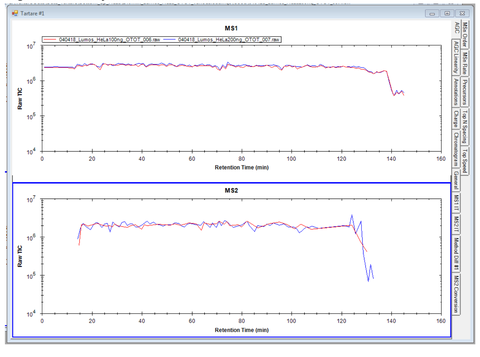
- General raw file information table
- Spectral counts plot
  - Supports MS^n
- Precursor charge state distributions
- Top N spacing
  - Separate plot for Top Speed methods
- Precursor intensity distributions
- Injection time plots
- Lock mass tracking
- Metadata time effects
  - Supports all parameters and includes spectral filtering
- Method Report and Error Log displays

NEW FEATURES

AGC Behavior

With trapped ion instruments, it's critical that the ion population be properly controlled in order to obtain reproducible results. The normal AGC intensity scaling can be backed out of the data to provide an indication of the number of ions in each spectrum.

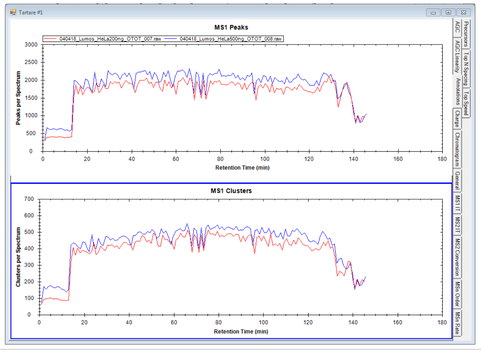
Figure 2. AGC Analysis of MS and MS2 data.



Annotations

For any data dependent experiment, the first step is acquiring a master scan and identifying peaks of interest for analysis by MS/MS. The number of peaks, and more importantly the number of isotopic clusters identified in the master scan, play a significant role in the ability to keep the instrument busy collecting MS/MS data.

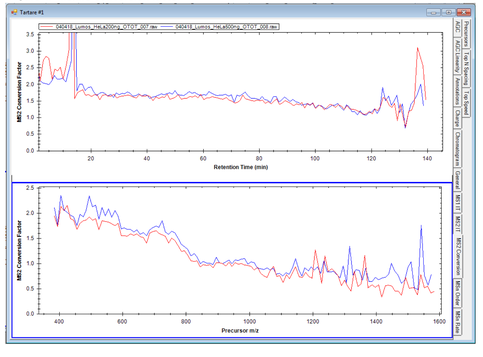
Figure 3. Analysis of Total Peaks and Clusters in MS1 Spectra.



MS^n Conversion Factors

The quality of MS/MS data is directly related to efficiency of both the isolation and activation steps. To evaluate this, one needs to compare the total signal observed in the MS/MS spectrum to the amount of signal in the expected isolation window of the corresponding master spectrum.

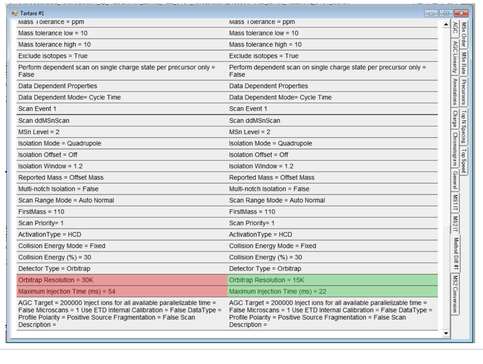
Figure 4. MS/MS conversion factor



Method Report Diff

Quality differences between two raw files can often be traced to differences in the method settings. However it's quite tedious to manually examine all the settings in order to identify differences between the methods. Tartare includes a text "Diff", which automatically highlights differences between methods.

Figure 5. Highlighted differences between methods.



Mass Tolerance = ppm	Mass Tolerance = ppm
Mass tolerance low = 10	Mass tolerance low = 10
Mass tolerance high = 10	Mass tolerance high = 10
Exclude isotopes = True	Exclude isotopes = True
Perform dependent scan on single charge state per precursor only = False	Perform dependent scan on single charge state per precursor only = False
Data Dependent Properties	Data Dependent Properties
Data Dependent Mode= Cycle Time	Data Dependent Mode= Cycle Time
Scan Event 1	Scan Event 1
Scan dsMSnScan	Scan dsMSnScan
MSn Level = 2	MSn Level = 2
Isolation Mode = Quadrupole	Isolation Mode = Quadrupole
Isolation Offset = Off	Isolation Offset = Off
Isolation Window = 1.2	Isolation Window = 1.2
Reported Mass = Offset Mass	Reported Mass = Offset Mass
Multi-notch Isolation = False	Multi-notch Isolation = False
Scan Range Mode = Auto Normal	Scan Range Mode = Auto Normal
FirstMass = 110	FirstMass = 110
Scan Priority= 1	Scan Priority= 1
ActivationType = HCD	ActivationType = HCD
Collision Energy Mode = Fixed	Collision Energy Mode = Fixed
Collision Energy (%) = 30	Collision Energy (%) = 30
Detector Type = Orbitrap	Detector Type = Orbitrap
Orbitrap Resolution = 30K	Orbitrap Resolution = 15K
Maximum Injection Time (ms) = 54	Maximum Injection Time (ms) = 22
AGC Target = 200000 Inject ions for all available parallelizable time = False Microscans = 1 Use ETD internal Calibration = False DataType = Profile Polarity = Positive Source Fragmentation = False Scan Description =	AGC Target = 200000 Inject ions for all available parallelizable time = False Microscans = 1 Use ETD internal Calibration = False DataType = Profile Polarity = Positive Source Fragmentation = False Scan Description =

Custom Analysis

Any user with limited programming skills can easily add custom analyses. For the simplest case, this requires the definition of a single Lua table which includes processFile() and generateReport() functions. If the custom analysis requires spectral data, then two additional functions are required which define the spectra that need to be processed, along with the actual processing.

Figure 6. Lua code listing for a simple analysis

```lua
-- Local variables
local msnOrder = {name = "MSn Order"}
local allResults = {}
local toolTip = [[Number of Spectra for Each MSn Order]]

function msnOrder.generateReport(notebook)
  if #allResults == 0 then return end
  local orderPane = zPane()
  local thisPage = multiPlotPage{name = "MSn Order", panes = {orderPane}}
  thisPage.pageControl.ToolTipText = toolTip
  local paneControl = orderPane.paneControl
  paneControl.XAxis.Title.Text = "MS Order"
  paneControl.YAxis.Title.Text = "Count"
  paneControl.Title.Text = "MSn Order"
  notebook:AddPage(thisPage)
  Application.DoEvents()                    -- Let windows draw the page

  tartare.histogram{pane = orderPane, data = allResults, key = "order", seriesType = "bar",
                    integer = true}
  barItemType.CreateBarLabels(orderPane.paneControl, false, "")
  orderPane.plotControl:AxisChange()
  orderPane.plotControl:Invalidate()

end

function msnOrder.processFile(rawFile, rawFileName, firstFile)
  if firstFile then allResults = {} end
  -- Set up the result table for this raw file
  local thisResult = {fileName = rawFileName or string.format("File #%d", #allResults + 1)}
  -- Loop through scans and collect all necessary data
  for scanNumber = rawFile.FirstSpectrumNumber, rawFile.LastSpectrumNumber do
    table.insert(thisResult, {order = rawFile:GetMSOrder(scanNumber)})
  end
  if #thisResult > 0 then table.insert(allResults, thisResult) end
end

-- Register this report
tartare.register(msnOrder)
```

CONCLUSIONS

Tartare provides a comprehensive package for the analysis of data dependent experimental quality for all Xcalibur-based mass spectrometers. Custom analysis routines can be created by the user to support any unmet needs that are not supported by the base package.

REFERENCES

1. Xcalibur Workbench: A Lua Based Data Browser Michael W. Senko, Graeme McAlister and Derek J. Bailey, ASMS Conference on Mass Spectrometry and Allied Topics, San Antonio, TX, 2016.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the assistance of the Tribrid marketing and science groups for supplying example raw files.

TRADEMARKS/LICENSING

ThermoFisher SCIENTIFIC